数据集介绍

- GEO链接： https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE65624
- 芯片平台： GPL81, [MG_U74Av2] Affymetrix Murine Genome U74A Version 2 Array

样品列表：

```
             title        source_name_ch1
GSM1602122 jos1986 Control liver sample 1
GSM1602123 jos1987 Control liver sample 2
GSM1602124 jos1988 Control liver sample 3
GSM1602125 jos1983   LIRKO liver sample 1
GSM1602126 jos1984   LIRKO liver sample 2
GSM1602127 jos1985   LIRKO liver sample 3
```

文章链接是：Flavin-containing monooxygenase 3 as a potential player in diabetes-associated atherosclerosis. *Nat Commun* 2015 Apr 7;6:6498. PMID: 25849138

核心步骤

获取并且检查表达量矩阵

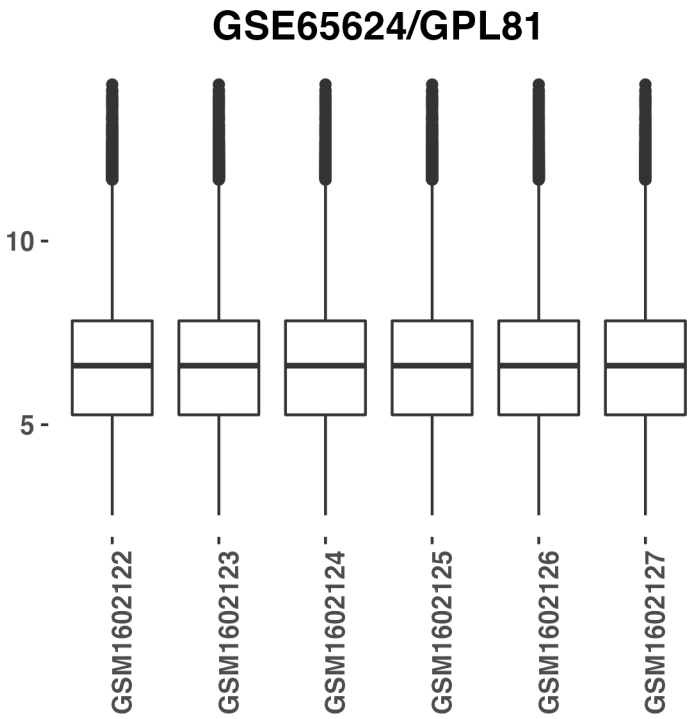
主要是得是否需要log

```
library(AnnoProbe)
library(GEOquery)
library(ggplot2)
library(ggstatsplot)
library(reshape2)

gse_number <- 'GSE65624'
gset <- geoChina(gse_number)
gset
gset[[1]]
a=gset[[1]]
dat=exprs(a) #a现在是一个对象，取a这个对象通过看说明书知道要用exprs这个函数
dim(dat)#看一下dat这个矩阵的维度
dat[1:4,1:4] #查看dat这个矩阵的1至4行和1至4列，逗号前为行，逗号后为列
boxplot(dat[,1:4],las=2)
# dat=log2(dat)
boxplot(dat[,1:4],las=2)
library(limma)
dat=normalizeBetweenArrays(dat)
#宽数据变长数据
class(dat)
```

```
data <- as.data.frame(dat)
data <- melt(data)
head(data)
title <- paste (gse_number, "/", a@annotation, sep ="")
p1 <- ggplot(data,aes(x=variable,y=value))+
      geom_boxplot()+
      theme_ggstatsplot()+
      theme(panel.grid = element_blank(),
            axis.text=element_text(size=10,face = 'bold'),
            axis.text.x=element_text(angle=90),
            plot.title = element_text(hjust = 0.5,size =15))+
      xlab('')+
      ylab('')+
      ggtitle(title)
p1
```

可以看到，处理前后我们的表达量矩阵的表达量范围箱线图如下所示：



根据生物学背景及研究目的人为分组

```
pd=pData(a)
#通过查看说明书知道取对象a里的临床信息用pData
##  挑选一些感兴趣的临床表型。
colnames(pd)
pd[,c(1,8)]
library(stringr)
group_list=ifelse(grepl('LIRKO',pd$source_name_ch1),'LIRKO','control')
table(group_list)
```

为了演示方便，我们这里仅仅是区分"LIRKO"和“control”。

获取芯片注释信息

代码如下：

```
ids=idmap('GPL81','soft')
```

可以看到此芯片的探针与基因ID或者symbol的对应关系如下所示：

```
> head(ids)
         ID symbol
1 100001_at   Cd3g
2 100002_at  Itih3
3 100003_at   Ryr1
4 100004_at  Ints7
5 100005_at  Traf4
6 100006_at  Cdh11
```

探针基因ID对应以及去冗余

代码如下：

```
ids=ids[ids$symbol != '',]
dat=dat[rownames(dat) %in% ids$ID,]
ids=ids[match(rownames(dat),ids$ID),]
head(ids)
colnames(ids)=c('probe_id','symbol')
ids$probe_id=as.character(ids$probe_id)
rownames(dat)=ids$probe_id
dat[1:4,1:4]

ids=ids[ids$probe_id %in%  rownames(dat),]
dat[1:4,1:4]
dat=dat[ids$probe_id,]

ids$median=apply(dat,1,median) #ids新建median这一列，列名为median，同时对dat这个矩阵按行操作，取每一行的中位数，将结果给到median这一列的每一行
ids=ids[order(ids$symbol,ids$median,decreasing = T),]#对ids$symbol按照ids$median中位数从大到小排列的顺序排序，将对应的行赋值为一个新的ids
ids=ids[!duplicated(ids$symbol),]#将symbol这一列取取出重复项，'!'为否，即取出不重复的项，去除重复的gene ，保留每个基因最大表达量结果s
dat=dat[ids$probe_id,] #新的ids取出probe_id这一列，将dat按照取出的这一列中的每一行组成一个新的dat
rownames(dat)=ids$symbol#把ids的symbol这一列中的每一行给dat作为dat的行名
dat[1:4,1:4]  #保留每个基因ID第一次出现的信息
```

最后得到了表达量矩阵如下所示：

```
> dat[1:4,1:4]  #保留每个基因ID第一次出现的信息
       GSM1602122 GSM1602123 GSM1602124 GSM1602125
Zzz3     6.133870   6.070264   6.366024   6.375640
Zyx      6.094047   6.277441   5.907205   6.120028
Zyg11b   6.002191   5.913411   6.046707   6.103834
Zxda     7.353415   7.344086   7.244835   7.276486
```

以及最简单的2分组，如下所示：

```
>table(group_list)
group_list
control   LIRKO
      3       3
```

保存为R数据文件：step1-output.Rdata

标准步骤之质控

需要出标准的**3张图**，包括主成分分析，高变基因的表达量热图，样品相关性热图

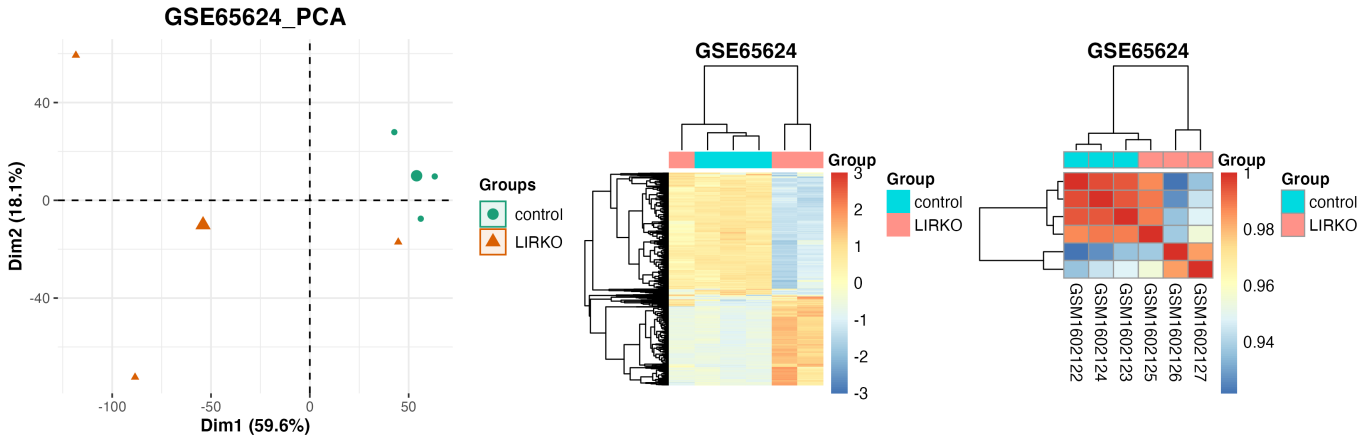
代码如下：

```
## 下面是画PCA的必须操作，需要看说明书。
exp=t(exp)#画PCA图时要求是行名时样本名，列名时探针名，因此此时需要转换
exp=as.data.frame(exp)#将matrix转换为data.frame
library("FactoMineR")#画主成分分析图需要加载这两个包
library("factoextra")
#主成分分析图p2
dat.pca <- PCA(exp , graph = FALSE)#现在exp最后一列是group_list，需要重新赋值给一个dat.pca,这个矩阵是不含有分组信息的
this_title <- paste0(gse_number,'_PCA')
p2 <- fviz_pca_ind(dat.pca,
                         geom.ind = "point", # show points only (nbut not "text")
                         col.ind = group_list, # color by groups
                         palette = "Dark2",
                         addEllipses = TRUE, # Concentration ellipses
                         legend.title = "Groups")+
                  ggtitle(this_title)+
                  theme_ggstatsplot()+
                  theme(plot.title = element_text(size=15,hjust = 0.5))

p2
#高变基因的表达量热图p3
cg=names(tail(sort(apply(dat,1,sd)),1000))#apply按行（'1'是按行取，'2'是按列取）取每一行的方差，从小到大排序，取最大的1000个
```

```
pheatmap(dat[cg,],show_colnames =F,show_rownames = F) #对那些提取出来的1000个基因所在的每一行取出，组合起来为一个新的表达矩阵
n=t(scale(t(dat[cg,]))) # 'scale'可以对log-ratio数值进行归一化
n[n>2]=2
n[n< -2]= -2
n[1:4,1:4]
pheatmap(n,show_colnames =F,show_rownames = F)
ac=data.frame(Group=group_list)
rownames(ac)=colnames(n)
p3 <- pheatmap::pheatmap(n,
                         show_colnames =F,
                         show_rownames = F,
                         main = gse_number,
                         annotation_col=ac,
                         breaks = seq(-3,3,length.out = 100))
p3
#样品相关性热图p4
colD=data.frame(Group=group_list)
rownames(colD)=colnames(exprSet)
p4 <- pheatmap::pheatmap(cor(exprSet),
                         annotation_col = colD,
                         show_rownames = F,
                         main = gse_number
                         )
p4
```

出图如下：



标准步骤之limma差异分析

代码如下：

```
library(limma)
design=model.matrix(~factor( group_list ))
fit=lmFit(dat,design)
fit=eBayes(fit)
options(digits = 4) #设置全局的数字有效位数为4
deg = topTable(fit,coef=2,adjust='BH', n=Inf)
```

差异分析结果前10行如下所示：

```
> deg[1:10,]
         logFC AveExpr      t   P.Value adj.P.Val     B
Lepr     5.358   6.723  40.05 1.655e-09 1.519e-05 7.718
Saa2    -4.112  11.023 -19.46 2.442e-07 1.120e-03 6.186
Cyp2b9   3.643   7.963  16.06 9.076e-07 2.097e-03 5.489
Cyp2b10  2.192   8.396  16.05 9.141e-07 2.097e-03 5.485
Igfbp2   1.890  12.337  15.08 1.397e-06 2.563e-03 5.229
Fmo3     4.471   7.503  14.54 1.789e-06 2.735e-03 5.074
Igfbp1   3.158  10.387  13.91 2.406e-06 3.153e-03 4.881
Gpx3     3.392   8.966  13.60 2.805e-06 3.217e-03 4.779
Akr1b7   1.847   7.350  12.82 4.178e-06 4.259e-03 4.504
Lcn2    -4.244   8.005 -12.43 5.134e-06 4.710e-03 4.357
```

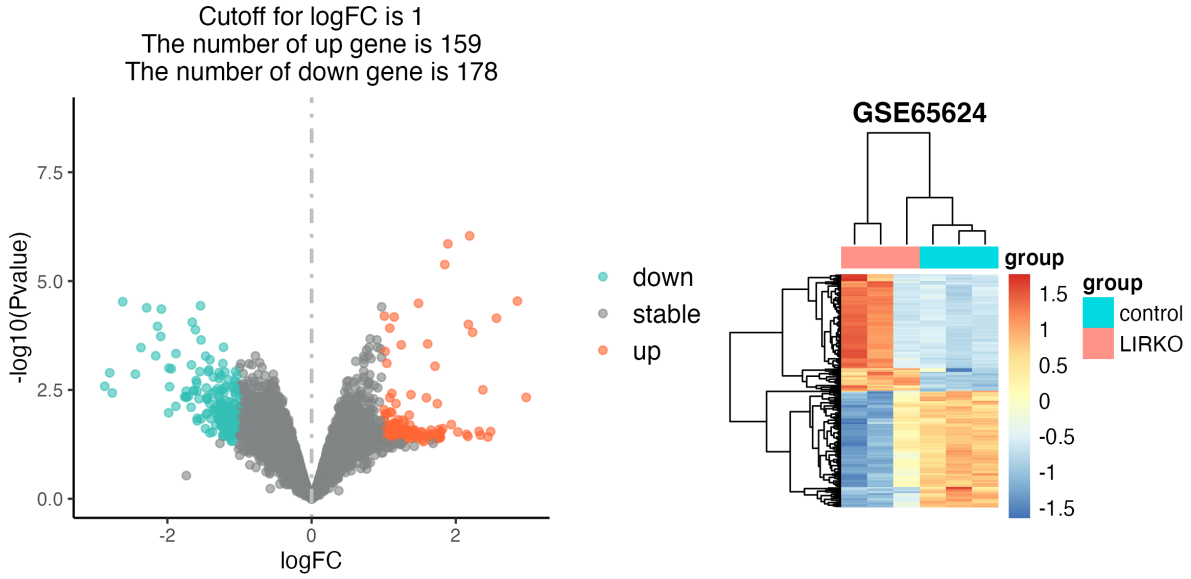
有了差异分析就可以进行标准的可视化，包括**火山图和上下调的差异基因热图**

代码如下：

```
nrDEG=deg
head(nrDEG)
attach(nrDEG)
plot(logFC,-log10(P.Value))
df=nrDEG
df$v= -log10(P.Value) #df新增加一列'v',值为-log10(P.Value)
df$g=ifelse(df$P.Value>0.05,'stable', #if 判断：如果这一基因的P.Value>0.01，则为stable基因
              ifelse( df$logFC >1,'up', #接上句else 否则：接下来开始判断那些P.Value<0.01的基因，再if 判断：如果logFC >1.5,则为up（上调）基因
                      ifelse( df$logFC < -1,'down','stable') )#接上句else 否则：接下来开始判断那些logFC <1.5 的基因，再if 判断：如果logFC <1.5，则为down（下调）基因，否则为stable基因
)
table(df$g)
df$name=rownames(df)
head(df)
logFC_t = 1
this_tile <- paste0('Cutoff for logFC is ',round(logFC_t,3),
                    '\nThe number of up gene is ',nrow(df[df$g == 'up',]) ,
                    '\nThe number of down gene is ',nrow(df[df$g == 'down',])
)
```

```
#火山图p5
p5 <- ggplot(data = df,
             aes(x = logFC,
                 y = -log10(P.Value))) +
    geom_point(alpha=0.6, size=1.5,
               aes(color=g)) +
    ylab("-log10(Pvalue)")+
    scale_color_manual(values=c("#34bfb5", "#828586","#ff6633"))+
    geom_vline(xintercept= 0,lty=4,col="grey",lwd=0.8) +
    xlim(-3, 3)+
    theme_classic()+
    ggtitle(this_tile )+
    theme(plot.title = element_text(size=12,hjust = 0.5),
          legend.title = element_blank(),
          )
p5
#上下调的差异基因热图p6
x=deg$logFC #deg取logFC这列并将其重新赋值给x
names(x)=rownames(deg) #deg取probe_id这列，并将其作为名字给x
cg=c(names(head(sort(x),100)),#对x进行从小到大排列，取前100及后100，并取其对应的探针名，作为向量赋值给cg
       names(tail(sort(x),100)))
library(pheatmap)
pheatmap(dat[cg,],show_colnames =F,show_rownames = F) #对dat按照cg取行，所得到的矩阵来画热图
n=t(scale(t(dat[cg,])))#通过“scale”对log-ratio数值进行归一化，现在的dat是行名为探针，列名为样本名，由于scale这个函数应用在不同组数据间存在差异时，需要行名为样本，因此需要用t(dat[cg,])来转换，最后再转换回来
n[n>2]=2
n[n< -2]= -2
n[1:4,1:4]
pheatmap(n,show_colnames =F,show_rownames = F)
ac=data.frame(group=group_list)
rownames(ac)=colnames(n) #将ac的行名也就分组信息（是‘no TNBC’还是‘TNBC’）给到n的列名，即热图中位于上方的分组信息
p6 <- pheatmap(n,show_colnames =F,
           show_rownames = F,
           cluster_cols = T,
           main = gse_number,
           annotation_col=ac) #列名注释信息为ac即分组信息
p6
```

出图如下：



标准步骤之生物学功能数据库注释

我们这里不根据**任何武断的阈值来区分统计学显著的上下调基因**，而是直接根据基因的变化情况排序进行gsea分析，而且仅仅是展示kegg这个生物学功能数据库的注释情况！

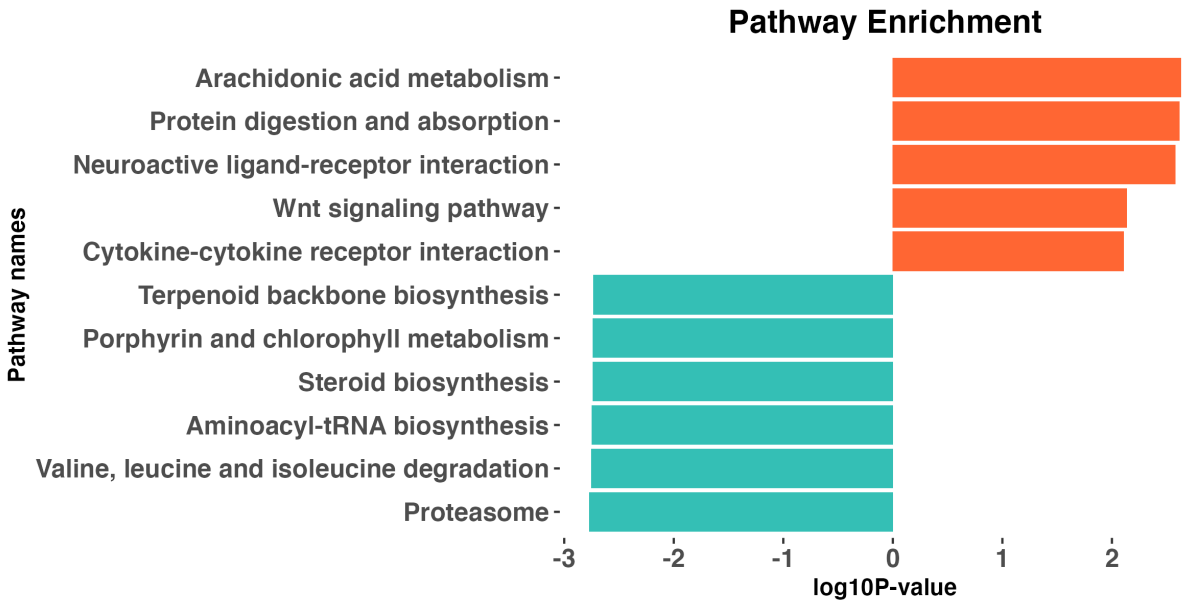
```
library(dplyr)
library(ggplot2)
geneList=DEG$logFC
names(geneList)=DEG$ENTREZID
geneList=sort(geneList,decreasing = T)
head(geneList)
library(clusterProfiler)
kk_gse <- gseKEGG(geneList     = geneList,
                  organism     = 'mmu',#按需替换
                  nPerm        = 1000,
                  minGSSize    = 10,
                  pvalueCutoff = 0.9,
                  verbose      = FALSE)
tmp=kk_gse@result
kk=DOSE::setReadable(kk_gse, OrgDb='org.Mm.eg.db',keyType='ENTREZID')#按需替换
tmp=kk@result
pro='comp1'
write.csv(kk@result,paste0(pro,'_kegg.gsea.csv'))
save(kk,file = 'gsea_kk.Rdata')
```

上面的kk这个变量就存储了kegg这个生物学功能数据库的gsea分析结果，我们进行简单可视化，代码如下：

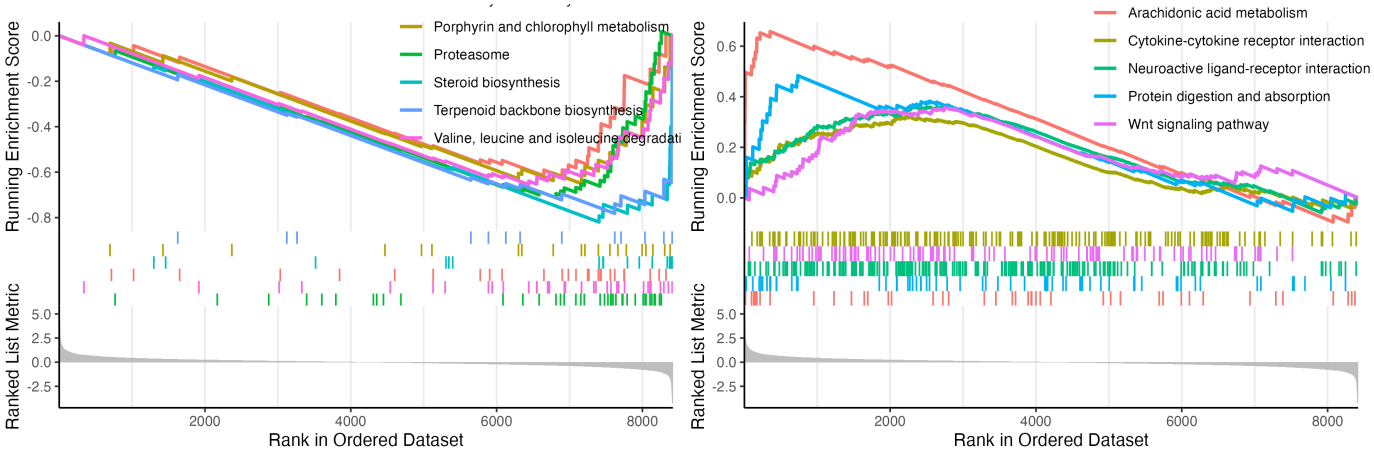
```
down_kegg<-kk_gse[kk_gse$pvalue<0.01 & kk_gse$enrichmentScore <
-0.6,];down_kegg$group=-1
up_kegg<-kk_gse[kk_gse$pvalue<0.01 & kk_gse$enrichmentScore > 0.3,];up_kegg$group=1
# 展现前6个上调通路和6个下调通路
dat=rbind(up_kegg,down_kegg)
```

```
colnames(dat)
dat$pvalue = -log10(dat$pvalue)
dat$pvalue=dat$pvalue*dat$group
dat=dat[order(dat$pvalue,decreasing = F),]
#gsea分析结果p7
p7<- ggplot(dat, aes(x=reorder(Description,order(pvalue, decreasing = F)), y=pvalue,
fill=group)) +
  geom_bar(stat="identity") +
  scale_fill_gradient(low="#34bfb5",high="#ff6633",guide = FALSE) +
  scale_x_discrete(name ="Pathway names") +
  scale_y_continuous(name ="log10P-value") +
  coord_flip() +
  theme_ggstatsplot()+
  theme(plot.title = element_text(size = 15,hjust = 0.5),
        axis.text = element_text(size = 12,face = 'bold'),
        panel.grid = element_blank())+
  ggtitle("Pathway Enrichment")
p7
#具体看上面条形图里面的每个通路的gsea分布情况p8
p8 <- gseaplot2(kk, geneSetID = rownames(down_k))+
      gseaplot2(kk, geneSetID = rownames(up_k))
p8
```

出图如下：



还可以具体看上面条形图里面的每个通路的gsea分布情况，如下所示：


Running Enrichment Score
0.0
-0.2
-0.4
-0.6
-0.8
Porphyrin and chlorophyll metabolism
Proteasome
Steroid biosynthesis
Terpenoid backbone biosynthesis
Valine, leucine and isoleucine degradati
Ranked List Metric
5.0
2.5
0.0
-2.5
2000
4000
6000
8000
Rank in Ordered Dataset
Running Enrichment Score
0.6
0.4
0.2
0.0
Arachidonic acid metabolism
Cytokine-cytokine receptor interaction
Neuroactive ligand-receptor interaction
Protein digestion and absorption
Wnt signaling pathway
Ranked List Metric
5.0
2.5
0.0
-2.5
2000
4000
6000
8000
Rank in Ordered Dataset